Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 381 (2007) 241–247

Theoretical Computer Science

www.elsevier.com/locate/tcs

Algorithms for minimum m-connected k-tuple dominating set problem[☆]

Weiping Shang[a,c,*], Pengjun Wan[b], Frances Yao[c], Xiaodong Hu[a]

[a] *Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing 10080, China*
[b] *Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA*
[c] *Department of Computer Science, City University of Hong Kong, Hong Kong*

Received 24 January 2007; received in revised form 25 April 2007; accepted 29 April 2007

Communicated by D.-Z. Du

Abstract

In wireless sensor networks, a virtual backbone has been proposed as the routing infrastructure to alleviate the broadcasting storm problem and perform some other tasks such as area monitoring. Previous work in this area has mainly focused on how to set up a small virtual backbone for high efficiency, which is modelled as the minimum Connected Dominating Set (CDS) problem. In this paper we consider how to establish a small virtual backbone to balance efficiency and fault tolerance. This problem can be formalized as the minimum m-connected k-tuple dominating set problem, which is a general version of minimum CDS problem with $m = 1$ and $k = 1$. We propose three centralized algorithms with small approximation ratios for small m and improve the current best results for small k.

Keywords: Connected dominating set; Approximation algorithm; k-vertex connectivity; Wireless sensor networks

1. Introduction

A Wireless Sensor Network (WSN) consists of wireless nodes (transceivers) without any underlying physical infrastructure. In order to enable data transmission in such networks, all the wireless nodes need to frequently flood control messages thus causing a lot of redundancy, contentions and collisions. To support various network functions such as multi-hop communication and area monitoring, some wireless nodes are selected to form a *virtual backbone*. Virtual backbone has been proposed as the routing infrastructure of WSNs. In many existing schemes (e.g., [1]) virtual backbone nodes form a Connected Dominating Set (CDS) of the WSN. With virtual backbones, routing messages are only exchanged between the backbone nodes, instead of being broadcasted to all the nodes. Prior work (e.g., [7]) has demonstrated that virtual backbones could dramatically reduce routing overhead.

☆ This work was supported in part by the Research Grants Council of Hong Kong under Grant No. CityU 1165/04E, the National Natural Science Foundation of China under Grant No. 70221001 and 10531070.

* Corresponding address: Academy of Mathematics and Systems Science, Chinese Academy of Science, No. 55, Zhongguancun east street, Beijing, China. Tel.: +852 68370454.

E-mail address: shangwp@amss.ac.cn (W. Shang).

doi:10.1016/j.tcs.2007.04.035

In WSNs, a node may fail due to accidental damage or energy depletion and a wireless link may fade away during node movement. Thus it is desirable to have several sensors monitor the same target, and let each sensor report via different routes to avoid losing an important event. Hence, how to construct a fault tolerant virtual backbone that continues to function when some nodes or links break down is an important research problem.

Recent work on sensor deployment and repairing [2,5] addresses the problems of deploying a sensor network from scratch or repairing a sensor network by adding new sensors to satisfy a certain connectivity requirement. In our study, we only need to choose a subset of nodes out of a pre-deployed network, instead of adding new nodes into the network.

In this paper we assume as usual that all nodes have the same transmission range (scaled to 1). Under such an assumption, a WSN can be modelled as a Unit Disk Graph (UDG) that consists of all nodes in the WSN and there exists an edge between two nodes if the distance between them is at most 1. We assume that the network is sufficiently dense such that the network is m-connected, and each node has at least k neighbors for a given k. The fault tolerant virtual backbone problem can be formalized as a combinatorial optimization problem: Given a UDG $G = (V, E)$ and two nonnegative integers m and k, find a subset of nodes $S \subseteq V$ of minimum size that satisfies: (i) each node u in V is *dominated* by at least k nodes in S (there is an edge between u and each of k nodes in S), (ii) S is *m-connected* (there are at least m disjoint paths between each pair of nodes in S). Every node in S is called a *backbone node* and every set S satisfying (i)–(ii) is called a *m-connected k-tuple dominating set* ((m, k)-CDS), and the problem is called a *minimum m-connected k-tuple dominating set* problem.

In this paper, we will study the minimum m-connected k-tuple dominating set problem for $m \leq 2$, which is important for the fault tolerant virtual backbone problem in WSNs. (When $m = 1$ and $k = 1$ the problem is reduced to the well known minimum connected dominating set problem.) We propose three centralized approximation algorithms to construct k-tuple dominating sets and m-connected k-tuple dominating sets for $m = 1, 2$. Our contributions in this paper are the following: (1) We propose new algorithms for $(1, k)$-CDS problems with performance ratio $(6+\ln \frac{5}{2}(k-1)+\frac{5}{k})$; In particular, an algorithm for (2, 1)-CDS problem with a performance ratio less than 24, which is better than the current best result. (2) We propose a centralized approximation algorithm for $(2, k)$-CDS problems for $k \geq 2$; To the best of our knowledge, this paper is the first to address this problem.

The remainder of this paper is organized as follows: Section 2 gives some definitions and Section 3 presents some related work. In Section 4 we present our algorithms with theoretical analysis on guaranteed performances. In Section 5 we conclude the paper.

2. Preliminaries

Let G be a graph with vertex-set $V(G)$ and edge-set $E(G)$. For any vertex $v \in V$, the neighborhood of v is defined by $N(v) \equiv \{u \in V(G) : uv \in E(G)\}$ and the closed neighborhood of v is defined by $N[v] \equiv \{u \in V(G) : uv \in E(G)\} \cup \{v\}$. The minimum degree of vertices in $V(G)$ is denoted by $\delta(G)$.

A subset $U \subseteq V$ is called an *independent set* (IS) of G if all vertices in U are pairwise non-adjacent, and it is further called a *maximal independent set* (MIS) if each vertex $V \setminus U$ is adjacent to at least one vertex in U.

A *dominating set* (DS) of a graph $G = (V, E)$ is a subset $S \subseteq V$ such that each vertex in $V \setminus S$ is adjacent to at least one vertex in S. A DS is called a *connected dominating set* (CDS) if it also induces a connected subgraph. A *k-tuple dominating set* (k-DS) $S \subseteq V$ of G is a set of vertices such that each vertex $u \in V$ is k-dominated by vertices of S. (Note that in some literatures, k-DS S only requires that each node in $V \setminus S$ is dominated by at least k nodes in S.)

A *cut-vertex* of a connected graph G is a vertex v such that the graph $G \setminus \{v\}$ is disconnected. A *block* is a maximal connected subgraph having no cut-vertex (so a graph is a block if and only if it is either 2-connected or equal to K_1 or K_2). The *block-cut-vertex graph* of G is a graph H where $V(H)$ consists of all cut-vertices of G and all blocks of G, with a cut-vertex v adjacent to a block G_0 if v is a vertex of G_0. The block-cut-vertex graph is always a forest. A *2-connected graph* is a graph without cut-vertices. Clearly a block with more than three nodes is a 2-connected component. A *leaf block* of a connected graph G is a block of G with only one cut-vertex.

3. Related work

Much effort has been made to design approximation algorithms for minimum connected dominating set problems. Wan et al. [9] proposed a two-phase distributed algorithm for the problem in UDGs that has a constant approximation performance ratio of 8. The algorithm first constructs a spanning tree, and then at the first phase, each node in a tree is examined to find a *Maximal Independent Set* (MIS) and all the nodes in the MIS are colored black. At the second phase, more nodes are added (color blue) to connect those black nodes. Recently, Li et al. [6] proposed another two-phase distributed algorithm with a better approximation ratio of $(4.8 + \ln 5)$. As in [9], at the first phase, an MIS is computed. At the second phase, a Steiner tree algorithm is used to connect nodes in the MIS. The Steiner tree algorithm is based on the property that any node in UDG is adjacent to at most 5 independent nodes.

The problem of minimum double domination and k-tuple domination sets has been studied in [3]. Ralf et al. [4] described a $(\ln |V| + 1)$-approximation algorithm for k-tuple dominating set problem in general graphs, and showed that this problem cannot be approximated within a ratio of $(1 - \varepsilon) \ln |V|$ for any $\varepsilon > 0$ unless $NP \subseteq DTIME(|V|^{O(\log\log|V|)})$. They also proposed an algorithm for the problem in UDGs that has an approximation ratio of $\frac{5}{2}(k - 1 + \frac{2}{k})$.

Most related to our work, Yang et al. [12] considered the problem of constructing the minimum $(1, k)$-CDS problem in WSNs. They proposed a cluster-based algorithm with performance ratio of $O(k^2)$ in unit disk graphs. Most recently, Wang et al. [10] proposed a 64-approximation algorithm for the minimum (2, 1)-CDS problem. The basic idea of this centralized algorithm is as follows: (i) Construct a small-sized CDS as a starting point of the backbone; (ii) iteratively augment the backbone by adding new nodes to connect a leaf block in the backbone to other block (or blocks); (iii) the augmentation process stops when all backbone nodes are in the same block, i.e., the backbone nodes are 2-connected. The augmentation process stops in at most $|\text{CDS}| - 1$ steps and in each step at most 8 nodes are added.

4. Approximation algorithms

Ralf et al. [4] proved the following lemma, which will be used in our performance analysis of proposed algorithms.

Lemma 1. *Let $G = (V, E)$ be a unit disk graph and k a constant such that $\delta(G) \geq k - 1$. Let D_k^* be a minimum k-tuple dominating set of G and S a maximal independent set of G. Then $|S| \leq \frac{5}{k}|D_k^*|$.*

4.1. Algorithm for computing k-DS and $(1, k)$-CDS

The basic idea of our algorithm for the minimum k-DS (i.e., $(1, k)$-CDS) problem is as follows: First, construct a set D by sequentially choosing an MIS k times (first choosing a CDS and then sequentially choosing an MIS $(k - 1)$ times) such that all vertices in $V \setminus D$ are k-dominated by set D; And then choose a set from $V \setminus D$ such that each vertex in D is also k-dominated. At each time we choose a vertex that can dominate the most number of vertices in S, where S is the vertex set that is not k-dominated up to now. The algorithm is more formally presented as follows.

Algorithm A for computing k-DS and $(1, k)$-CDS

```
Choose an MIS I_1 of G and a set C such that I_1 ∪ C is a CDS (refer to [9])
for i := 2 to k
    Construct an MIS I_i in G \ I_1 ∪ ··· ∪ I_{i-1}
end for
    D := I_1 ∪ ··· ∪ I_k and D_k := D
    m(v) := k − |N[v] ⋂ D_k| for each v ∈ D
    S = {v ∈ D : m(v) > 0}
    while S ≠ ∅ do
        Find vertex u ∈ V \ D_k that has the most neighbors in S
        D_k := D_k ∪ {u}
        If v ∈ N(u) ∩ S then m(v) := m(v) − 1
        S := {v ∈ D : m(v) > 0}
    end-while
return D_k and D_k^c = C ∪ D_k
```

Theorem 1. *Algorithm* A *returns a solution that is a $(6 + \ln \frac{5}{2}(k - 1))$-approximate solution to the minimum k-tuple dominating set problem, and also a $(6 + \ln \frac{5}{2}(k - 1) + \frac{5}{k})$-approximate solution to the minimum connected k-tuple dominating set problem.*

Proof. Suppose that Algorithm A, given graph $G = (V, E)$ and a natural number $k \geq 1$, returns D_k, $C \cup D_k$. Let D_k^* be a minimum k-tuple domination of G. We will show that D_k and $C \cup D_k$ are a k-tuple dominating set and connected k-tuple dominating set of G, respectively. For all $u \in G \setminus I_1 \cup \cdots \cup I_k$, at the i-th iteration, u is not in I_i and thus it is dominated by one vertex of I_i. At the end, u is dominated by at least k different vertices of $I_1 \cup \cdots \cup I_k$. When the algorithm terminates, for each $u \in I_1 \cup \cdots \cup I_k$, $m(u) \leq 0$, i.e., $|N[u] \bigcap D_k| \geq k$. Hence, u is k-dominated and D_k is a k-tuple dominating set.

By the first step of Algorithm A, $C \cup I_1$ is a CDS and thus $C \cup D_k$ is connected. By the rule of Algorithm A, we have $D = I_1 \cup \cdots \cup I_k$ and each I_i is an MIS. Thus it follows from Lemma 1 that $|I_i| \leq \frac{5}{k}|D_k^*|$. Hence we have $|D| \leq 5|D_k^*|$. In the following we will show that $|D_k \setminus D| \leq (1 + \ln \frac{5}{2}(k-1))|D_k^*|$.

Let $u_1, \ldots, u_l$ be the vertices in $D_k \setminus D$ in the order of their appearances in the algorithm. Then $D^i = D \cup \{u_1, u_2, \ldots, u_i\}$, and for any $v \in D$, $m^i(v) = k - |N[v] \bigcap D^i|$, $S^i = \{v \in D : m^i(v) > 0\}$, $M^i = \sum_{v \in D} m^i(v)$. Let C^i be an optimal solution such that each vertex in S^i is $m^i(v)$-dominated by C^i. Clearly, $|C^i| \leq |D_k^*|$. Note that for each $i = 1, 2, \ldots, k$ and each vertex $u \in I_i$, $|N[v] \cap D^0| \geq i$, where $D^0 = D$. Indeed, vertex u is i-dominated by itself and at least one vertex in I_j for each $1 \leq j \leq i-1$. Hence, $m^0(u) \leq k - i$. We have

$$M^0 = \sum_{v \in D} m^0(v) \leq \sum_{i=1}^{k} (k-i)|I_i| \leq \frac{5}{2}(k-1)|D_k^*|.$$

Since at the $(i+1)$-th iteration, we choose a vertex u such that u has the most neighbors in S^i, the number of neighbors of u in S^i must be at least $\frac{M^i}{|C^i|}$. Thus we have

$$M^{i+1} \leq M^i - \frac{M^i}{|C^i|} \leq M^i - \frac{M^i}{|D_k^*|}.$$

Note that $M^0 \leq \frac{5}{2}(k-1)|D_k^*|$ and $M^{l+1} = 0$. There exists a natural number $h \leq l$ such that $M^h \geq |D_k^*|$ and $M^{h+1} < |D_k^*|$. So,

$$M^h \leq M^0 \left(1 - \frac{1}{|D_k^*|}\right)^h \leq M^0 e^{\frac{h}{|D_k^*|}}.$$

Thus we have

$$\frac{h}{|D_k^*|} \leq \ln \frac{M^0}{M^h} \leq \ln \frac{5}{2}(k-1),$$

and then we obtain

$$l \leq h + M^{h+1} \leq \left(1 + \ln \frac{5}{2}(k-1)\right)|D_k^*|.$$

Hence, $|D_k \setminus D| \leq (1 + \ln \frac{5}{2}(k-1))|D_k^*|$.

In the end, let C be the set constructed from the first step of Algorithm A. By using the argument for the proof of Lemma 10 in [9], we can deduce $|C| \leq |I_1|$. Hence it follows from Lemma 1 that $|C| \leq \frac{5}{k}|D_k^*|$, and the size of connected k-tuple dominating set D_k^c is bounded by $(6 + \ln \frac{5}{2}(k-1) + \frac{5}{k})|D_k^*|$ and the size of the optimal solution of connected k-tuple dominating set is at least $|D_k^*|$. The proof is then finished. □

4.2. *Algorithm for computing* $(2, k)$*-CDS*

The basic idea of our algorithm for the minimum $(2, k)$-CDS problem with $k \geq 2$ is similar to the method proposed in [10]. It essentially consists of the following four steps:

Step 1. Apply Algorithm A to construct a connected k-tuple dominating set D.
Step 2. Compute all the blocks in D by computing the 2-connected components through the depth first search.

Step 3. Produce the shortest path in the original graph such that it can connect a leaf block in D with other part of D but does not contain any vertices in D except the two endpoints. Then add all intermediate vertices in this path to D.

Step 4. Repeat Step 2 and Step 3 until D is 2-connected.

In Step 2, we can apply the standard algorithm proposed in [8] to compute all blocks in D, denote the number of blocks in D by ComputeBlock (D). The algorithm is more formally presented as follows:

Algorithm B for computing a 2-connected k-tuple dominating set ($k \geq 2$)

```
1. Choose a connected k-tuple dominating set D_k^c using Algorithm A
2. D := D_k^c and B := ComputeBlocks(D)
3. while B > 1 do
4.      Choose a leaf block L
5.      for vertex v ∈ L not a cut-vertex do
6.          for vertex u ∈ V \ L do
7.              Construct G' from G by deleting all nodes in D except u and v
8.              P_uv :=shortestPath(G'; v, u) and P := P ∪ P_uv
9.          end-for
10.     end-for
11.     P_{u0v0} := the shortest path in P
12.     D := D∪ the intermediate vertices in P_{u0v0}
13.     ComputeBlocks(D)
14. end-while
15. return D
```

Lemma 2. *For $k \geq 2$, at most two new vertices are added into D at each augmenting step.*

Proof. Suppose that L is a leaf block of D at one augmenting step and w is the cut-vertex that is adjacent to the block L. where we assume that $w \in L$. Suppose vertices $u_0 \in L \setminus \{w\}$ and $v_0 \in V \setminus L$ are the two vertices connected by the shortest possible path $P_{u_0v_0}$ returned by the Algorithm in Step 12. We claim that $P_{u_0v_0}$ has at most two intermediate vertices. Suppose, by contradiction, that $P_{u_0v_0}$ contains $u_0, x_1, x_2, \ldots, x_l, v_0$, where $l \geq 3$, and $x_i \notin D$ for $i = 1, 2, \ldots, l$. Since D is a k-tuple dominating set for $k \geq 2$, each vertex x_i has at least 2 neighbors in D, and $N(x_i) \cap D \subseteq L$ or $N(x_i) \cap D \subseteq (V \setminus L) \cup \{w\}$. Otherwise, let $u_i \in L \setminus \{w\}$ and $v_i \in V \setminus L$ be the two neighbors of x_i, then the path $P_{u_iv_i}$ between u_iv_i has a shorter distance than $P_{u_0v_0}$. And it is clear that $N(x_1) \cap D \subseteq L$, $N(x_l) \cap D \subseteq (V \setminus L) \cup \{w\}$. If $N(x_2) \cap D \subseteq L$, x_2 must have a neighbor s in $L \setminus \{w\}$, then the path between sv_0 has a shorter distance than $P_{u_0v_0}$. Otherwise $N(x_2) \bigcap D \subseteq (V \setminus L) \cup \{w\}$, x_2 must have a neighbor s in $V \setminus L$, then the path between u_0s has a shorter distance than $P_{u_0v_0}$. Which contradicts that $P_{u_0v_0}$ has the shortest distance. □

Lemma 3. *The number of cut-vertices in the connected k-tuple dominating set D_k^c by Algorithm* A *is no bigger than the number of connected dominating sets in $I_1 \cup C$ chosen in Step* 1 *of Algorithm* A.

Proof. Let $S = I_1 \cup C$ be a connected domination set. We will show that no vertex in $D_k^c \setminus S$ is a cut-vertex. For any two vertices $u, v \in S$, there is a path P_{uv} between them that contains only vertices in S. Since any vertex in $D_k^c \setminus S$ is dominated by at least one vertex in S, Hence, for any two vertices $u, v \in D_k^c$, there is a path P_{uv} between them that contains only vertices in $S \bigcup \{u, v\}$. Hence, any vertex in $D_k^c \setminus S$ is not a cut-vertex. □

Theorem 2. *Algorithm* B *returns a* $(6 + \ln \frac{5}{2}(k-1) + \frac{25}{k})$*-approximate solution to the minimum 2-connected k-tuple dominating set problem for $k \geq 2$.*

Proof. Let D_k^* and D_{opt} be the optimal k-tuple dominating set and 2-connected k-tuple dominating set, respectively. It is clear that $|D_k^*| \leq |D_{\text{opt}}|$. After S is constructed, by Lemmas 2 and 3, the algorithm terminates in at most $|C| + |I_1|$ steps, and in each step at most two vertices are added. Since $|C| + |I_1| \leq 2|I_1| \leq \frac{10}{k}|D_k^*|$, we have $|D| \leq |D_k^c| + \frac{20}{k}|D_k^*|$. It follows from Theorem 1 that $|D_k^c| \leq (6 + \ln \frac{5}{2}(k-1) + \frac{5}{k})|D_k^*|$. Hence we obtain $|D| \leq (6 + \ln \frac{5}{2}(k-1) + \frac{25}{k})|D_{\text{opt}}|$. □

4.3. *Algorithm for computing* (2, 1)*-CDS*

Lemma 2 is true when $k \geq 2$. If Algorithm B is used for the case where $k = 1$, then more vertices are added into D at each augmenting step. In order to design an algorithm with a smaller performance ratio, we propose Algorithm C. The main idea of this algorithm is as follows: First, construct a connected dominating set C using the algorithm in [6], and then construct a maximal independent set D in $G \setminus C$, in the end make $C \cup D$ to be 2-connected by adding some new vertices to it.

Algorithm C for computing 2-connected dominating set

1. Produce a connected dominating set C of G using the algorithm in [6].
2. Construct a maximal independent set D in $G \setminus C$
3. $S := C \cup D$
4. Augment S using Steps 2-14 of Algorithm B

Theorem 3. *Algorithm C returns a 2-connected dominating set whose size is at most* $(18.2 + 3\ln 5)|D_2^*| + 4.8$*, where* $|D_2^*|$ *is the size of the optimal 2-connected dominating set.*

Proof. Let D_1^* and D_2^* be the optimal (1, 1)-CDS and (2, 1)-CDS, respectively. It is clear that $|D_1^*| \leq |D_2^*|$. After C and D are constructed, which are a connected dominating set of G and a dominating set of $G \setminus C$, respectively, each vertex in $V \setminus S$ is dominated by at least two vertices in S. Thus, Lemmas 2 and 3 also hold true for Algorithm C. In addition, it follows from Lemmas 2 and 3 that at most $|C|$ steps are needed before the algorithm terminates, and at each step at most two vertices are added. Hence, we obtain $|S| \leq 3|C| + |D|$. Using the same argument for Theorem 1 in [6,11], we could show $|C| \leq (4.8 + \ln 5)|D_1^*| + 1.2$ and $|D| \leq 3.8|D_1^*| + 1.2$ respectively. Thus we obtain $|S| \leq (18.2 + 3\ln 5)|D_2^*| + 4.8$. □

Observe that $(18.2 + 3\ln 5) < 23.03$. So Algorithm C has a better guaranteed performance than the 64-approximation algorithm in [10] for the same problem (when the size of the optimal 2-connected dominating set is not very big).

5. Conclusion

In this paper we have proposed three centralized approximation algorithms for the minimum k-tuple dominating set problem and m-connected k-tuple dominating set problem for $m = 1, 2$. Although the approximation performance ratios of Algorithms A and B are dependent on k, they are very small when k is not very big, in fact, that is the case for virtual backbone construction in wireless sensor networks. Our future work is to extend our study to the more general case of $m \geq 3$, and design distributed and localized algorithms for the minimum m-connected k-tuple dominating set problem.

Acknowledgment

The authors are deeply indebted to anonymous referees for their invaluable comments and suggestions which have greatly improved the presentation of this paper.

References

[1] K.M. Alzoubi, P.-J. Wan, O. Frieder, Distributed heuristics for connected dominating sets in wireless ad hoc networks, Journal of Communications and Networks 4 (1) (2002) 22–29.
[2] J.L. Bredin, E.D. Demaine, M. Hajiaghayi, D. Rus, Deploying sensor networks with guaranteed capacity and fault tolerance, in: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc, 2005, pp. 309–319.
[3] F. Harary, T. Haynes, Double domination in graphs, ARS Combinatoria 55 (1) (2000) 201–213.
[4] R. Klasing, C. Laforest, Hardness results and approximation algorithms of k-tuple domination in graphs, Information Processing Letters 89 (2) (2004) 75–83.
[5] H. Koskinen, J. Karvo, O. Apilo, On improving connectivity of static ad-hoc networks by adding nodes, in: Proceedings of the 4th annual Mediterranean Workshop on Ad Hoc Networks, Med-Hoc-Net, 2005, pp. 169–178.
[6] Y.S. Li, M.T. Thai, F. Wang, C.-W. Yi, P.-J. Wan, D.-Z. Du, On greedy construction of connected dominating sets in wireless networks, Wiley Journal on Wireless Communications and Mobile Computing 5 (8) (2005) 927–932.

[7] P. Sinha, R. Sivakumar, V. Bharghavan, Enhancing ad hoc routing with dynamic virtual infrastructures, in: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, 2001, pp. 1763–1772.
[8] R. Tarjan, Depth first search and linear graph algorithms, SIAM Journal on Computing 1 (2) (1972) 146–160.
[9] P.-J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, Mobile Networks and Applications 9 (2) (2004) 141–149.
[10] F. Wang, T. Thai, D.Z. Du, On the construction of 2-connected virtual backbone in wireless networks, IEEE Transactions on Wireless Communications (in press).
[11] W. Wu, H. Du, X. Jia, Y. Li, C.-H. Huang, Minimum connected dominating sets and maximal independent sets in unit disk graphs, Theoretical Computer Science 352 (1) (2006) 1–7.
[12] S. Yang, F. Dai, M. Cardei, J. Wu, On multiple point coverage in wireless sensor networks, in: Proc. of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS, Nov. 2005.